Granular Computing
https://doi.org/10.1007/s41066-017-0065-2

ORIGINAL ARTICLE

Multi-task learning for intelligent data processing in granular computing context

Han Liu[1] · **Mihaela Cocea**[2] · **Weili Ding**[3]

Received: 24 October 2017 / Accepted: 15 November 2017

Abstract
Classification is a popular task in many application areas, such as decision making, rating, sentiment analysis and pattern recognition. In the recent years, due to the vast and rapid increase in the size of data, classification has been mainly undertaken in the way of supervised machine learning. In this context, a classification task involves data labelling, feature extraction, feature selection and learning of classifiers. In traditional machine learning, data is usually single-labelled by experts, i.e., each instance is only assigned one class label, since experts assume that different classes are mutually exclusive and each instance is clear-cut. However, the above assumption does not always hold in real applications. For example, in the context of emotion detection, there could be more than one emotion identified from the same person. On the other hand, feature selection has typically been done by evaluating feature subsets in terms of their relevance to all the classes. However, it is possible that a feature is only relevant to one class, but is irrelevant to all the other classes. Based on the above argumentation on data labelling and feature selection, we propose in this paper a framework of multi-task learning. In particular, we consider traditional machine learning to be single task learning, and argue the necessity to turn it into multi-task learning to allow an instance to belong to more than one class (i.e., multi-task classification) and to achieve class specific feature selection (i.e., multi-task feature selection). Moreover, we report two experimental studies in terms of fuzzy multi-task classification and rule learning based multi-task feature selection. The results show empirically that it is necessary to undertake multi-task learning for both classification and feature selection.

Keywords Machine learning · Multi-task learning · Image processing · Fuzzy classification · Granular computing

1 Introduction

Classification is one of the most popular tasks of machine learning, which has been frequently used in broad application areas, such as decision making (Pedrycz and Chen 2015a; Liu and Gegov 2015), sentiment analysis (Pedrycz and Chen 2016; Liu et al. 2016b) and pattern recognition (Teng et al. 2007). In general, classification is aimed at assigning a class/label to an unseen instance, i.e., it is to judge to which category an instance belongs.

In traditional machine learning, classification is typically considered to be of single-task, due to the following aspects:

Firstly, classification is generally undertaken by assuming that different classes are mutually exclusive and thus an instance can only belong to one class. However, this assumption does not really hold for many real-life problems. For example, in the context of text classification, the same movie may belong to different categories. Similarly, the same book may belong to different subjects. There are

✉ Han Liu
LiuH48@cardiff.ac.uk

Mihaela Cocea
mihaela.cocea@port.ac.uk

Weili Ding
weiye51@ysu.edu.cn

[1] School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, UK

[2] School of Computing, University of Portsmouth, Buckingham Building, Lion Terrace, Portsmouth PO1 3HE, UK

[3] Department of Automation, Institute of Electrical Engineering, Yanshan University, 438 West of Hebei Avenue, Haigang District, Qinghuangdao 066004, People's Republic of China

Published online: 30 November 2017

also many similar examples in other areas, e.g., a patient may be found to have more than one health issue in medical diagnosis.

Secondly, feature evaluation and selection have been considered as a very important step towards advancing the performance of learning classifiers (Liu et al. 2017a; Dash and Liu 1997; Langley 1994). However, the evaluation of features has been typically done by measuring their relevance to all classes. In fact, it could happen that a feature is only relevant to one class and is irrelevant to all the other classes (Cendrowska 1987). For example, in the context of image understanding, some target regions need to be identified, and each of the target regions involves recognizing instances of a specific class and extracting a set of features (that may be relevant only to this class). In this case, if features extracted from different target regions of an image are put together to make up a feature set, then the resulted data set could involve a sparse matrix. The case of a sparse matrix could lead to a large number of features being judged as irrelevant and thus filtered. However, these filtered features may be highly relevant to a specific class, which may lead to poor classification performance on that particular class.

Based on the two aspects described above, we argue in this paper the need to turn single-task learning into multi-task learning, i.e., a learning task per class. In particular, we propose the use of fuzzy approaches to allow an instance to belong to more than one class by judging the membership degree of an instance to different classes. We also show how different classes may be related to each other from granular computing perspective, through looking at fuzzy membership degrees of an instance to different classes, i.e., each class is viewed as a granule and the possible relationships between granules are identified.

On the other hand, in terms of feature evaluation and selection, we propose to turn it into a multi-task approach, from a granular computing perspective. In particular, we transform the class feature into a number of binary features, and each binary feature corresponds to a class. In this way, features are evaluated for each class in terms of their relevance, i.e., for each class, there is a feature subset selected towards learning to judge if an instance belongs to this class or not. We also show how rule learning approaches are capable of achieving class-specific feature evaluation.

The rest of this paper is organized as follows: Sect. 2 provides related work on classification, feature selection and granular computing. In Sect. 3, we present how single-task learning can be transformed effectively into multi-task learning, in terms of both classification and feature selection. In Sect. 4, we conduct two experimental studies, and discuss the results for showing the necessity to achieve multi-task classification and feature selection, towards advancing machine learning techniques for classification. In Sect. 5, we summarize the contributions of this paper and suggest further directions that could lead to advances in this research area in the future.

2 Related work

In this section, we describe the concepts of granular computing and justify how granular computing is related to classification and feature selection. Moreover, we provide an overview of classification in the context of machine learning and a review of existing approaches of feature selection.

2.1 Granular computing

Granular computing is a computational approach of information processing. It is aimed at structural thinking at the philosophical level and is aimed at structural problem-solving at the practical level (Yao 2005b). In general, granular computing involves two operations, namely granulation and organization (Yao 2005a). The former operation is to decompose a whole into parts, whereas the latter operation is to integrate parts into a whole. In computer science, granulation and organization have been frequently involved as the top-down and bottom-up approaches, respectively (Liu and Cocea 2017a).

In practice, two main concepts of granular computing, which have been popularly used for granulation and organization, are granule and granularity. A granule generally represents a large particle, which consists of smaller particles that can form a larger unit. There are many real-life examples as follows:

- In the context of classification, each class can be viewed as a granule, since a class represents a collection of objects/instances.
- In the context of feature selection, each feature set can be viewed as a granule, since a feature set represents a collection of features.

In general, granules can be at the same level or different levels with specific interrelationships, which leads to the need of the concept of granularity (Pedrycz and Chen 2015b). In particular, if granules are located at the same level of granularity, then the relationships between these granules are referred to as horizontal relationships (Liu and Cocea 2018). In contrast, for granules located at different levels of granularity, the relationships between these granules are referred to as hierarchical relationships (Liu and Cocea 2018). For example, in the context of classification, a class at a higher level of granularity may be specialized/decomposed into sub-classes at a lower level of granularity, in terms of specialization/decomposition (hierarchical relationships). Also, classes at a lower level of granularity may be generalized/aggregated into a super class at a higher

level of granularity, in terms of generalization/aggregation (hierarchical relationships) (Liu and Cocea 2017a). On the other hand, classes may also have horizontal relationships between each other when these classes are at the same level of granularity, such as mutual exclusion, correlation and mutual independence (Liu et al. 2017b).

In practice, granular computing concepts and techniques have been used broadly in popular areas, such as artificial intelligence (Wilke and Portmann 2016; Pedrycz and Chen 2011; Skowron et al. 2016), computational intelligence (Dubois and Prade 2016; Yao 2005b; Kreinovich 2016; Livi and Sadeghian 2016), machine learning (Min and Xu 2016; Peters and Weber 2016; Liu and Cocea 2017c; Antonelli et al. 2016), decision making (Xu and Wang 2016; Liu and You 2017; Chatterjee and Kar 2017) and data clustering (Chen et al. 2009; Horng et al. 2005).

Furthermore, ensemble learning is also a subject that involves applications of granular computing concepts (Liu and Cocea 2017c). In particular, ensemble learning approaches, such as Bagging, involve information granulation through decomposing a training set into a number of overlapping samples, and also involve organization through combining the predictions provided from different base classifiers towards classifying an unseen instance; a similar perspective has also been stressed and discussed in Hu and Shi (2009).

In Sect. 3, we will show how granular computing concepts can be used for advancing classification and feature selection in the context of multi-task learning.

2.2 Overview of classification

As mentioned in Sect. 1, classification is one of the most popular tasks of machine learning. In terms of the number of predefined classes for a learning task, classification can be specialized into two categories: binary classification and multi-class classification. On the other hand, classification can be for different purposes, which leads to different types of class attributes, such as nominal, ordinal and string (Tan et al. 2005). In this context, the purposes of classification include recognition, rating and decision making.

Both binary classification and multi-class classification tasks could essentially be for any of the above purposes. In particular, binary classification could be for the purpose of recognition, such as gender classification (Wu et al. 2011). There are also some examples of binary classification for the purpose of rating, such as sentiment analysis (positive or negative) and assessment of teaching and learning (good or bad). In addition, binary classification can be involved in a decision-making task, such as voting (support or objection) and shopping (buy or not). Regarding multi-class classification, examples of recognition include emotion identification (Teng et al. 2007; Altrabsheh et al. 2015). There are also many examples of rating, such as movie rating and multi-sentiment analysis (Jefferson et al. 2017). In addition, multi-class classification can be used as a way of decision making towards selecting one of the given options.

As argued in Sect. 1, in traditional machine learning, different classes are assumed to be mutually exclusive, but this assumption does not always hold in reality. To address this issue, some related work was done in Boutell et al. (2004); Tsoumakas and Katakis (2007); Tsoumakas et al. (2010); Zhang and Zhou (2014) for turning single-label classification into multi-label classification. In particular, multi-label classification typically includes three types: PT3, PT4 and PT5 (Tsoumakas and Katakis 2007).

PT3 is designed to enable that a class consists of two or more labels as illustrated in Table 1. For example, two classes A and B can make up three labels: A, B and $A \wedge B$. PT4 is designed to do the labelling on the same dataset separately regarding each of the predefined labels as illustrated in Tables 2 and 3. In addition, PT5 is aimed at uncertainty handling. In other words, it is not certain to which class label an instance should belong, so the instance is assigned all the possible labels and is treated as several different instances that have the same inputs but different class labels assigned. An illustrative example is given in Table 4: both instances (3 and 4) appear twice with two different labels (A and B) respectively, which would be treated as four different instances (two assigned A and the other two assigned B) in the process of learning.

As mentioned in Sect. 2.1, there could be different types of relationships between classes. From this point of view, multi-label classification approaches still have limitations

Table 1 Example of PT3 (Liu et al. 2017b; Liu and Cocea 2018)

Instance ID	Class
1	A
2	B
3	$A \wedge B$
4	$A \wedge B$

Table 2 Example of PT4 on Label A (Liu et al. 2017b; Liu and Cocea 2018)

Instance ID	Class
1	A
2	$\neg A$
3	A
4	A

Table 3 Example of PT4 on label B (Liu et al. 2017b; Liu and Cocea 2018)

Instance ID	Class
1	$\neg B$
2	B
3	B
4	B

Table 4 Example of PT5 (Liu et al. 2017b; Liu and Cocea 2018)

Instance ID	Class
1	*A*
2	*B*
3	*A*
3	*B*
4	*A*
4	*B*

as argued in Liu et al. (2017b), and Liu and Cocea (2018) as follows:

PT3 may result in a massive number of classes, i.e., $2^n - 1$, where n is the number of class labels. Also, PT3 may result in high coupling from software engineering perspective, while different class labels are not correlated but are merged into a new class. Coupling generally refers to the degree of interdependence between different parts (Lethbridge and Laganire 2005).

PT4 may result in the class imbalance issue. For example, a balanced dataset contains instances of three classes A, B and C, and thus the frequency $\left(\frac{1}{3}\right)$ of class A is far lower than the one $\left(\frac{2}{3}\right)$ of class ¬A (i.e. $B \vee C$). From software engineering perspective, PT4 may also result in low cohesion, while different class labels that are correlated get separated. Low cohesion means the degree to which the parts of a whole link together is lower (Lethbridge and Laganire 2005), and thus failing to identify the correlations between different classes.

PT5 may result in a massive size of training sample leading to high computational complexity, especially when the number of class labels is high in the big data era. Also, from machine learning perspective, PT5 may result in confusion for a learning algorithms. In other words, when a training set contains instances that have the same input vector but are assigned different class labels, the initial uncertainty in the dataset would be increased leading to the difficulty in discriminating between classes in the training process, since popular learning methods typically belong to discriminative learning.

Overall, the above multi-label classification approaches still aim at classifying uniquely a test instance, towards assigning the instance a single class, although this class may consist of more than one label. From granular computing perspective, single-label classification is aimed at providing a string as the output, whereas multi-label classification is aimed at providing a list of strings (as a whole) as the output. However, there is no fundamental difference between the two ways of classification in terms of the strategy of learning classifiers, i.e., discriminative learning. On the basis of the above argumentation, we consider both single-label classification and multi-label classification to be of single-task, and a framework of multi-task classification will be presented in Sect. 3.

2.3 Review of feature selection techniques

As introduced in Dash and Liu (1997), the feature selection process typically involves four main steps: generation, evaluation, stopping criterion and validation. In particular, the generation procedure is aimed at generating a candidate feature subset based on the original feature set. In the evaluation stage, a function is used to evaluate the goodness of the feature subset selected in the generation stage, in terms of importance of these selected features. A stopping criterion is then used to decide whether it is necessary to stop the feature selection process. If yes, the selected feature subset is validated in the last stage. Otherwise, the feature selection process needs to be repeated through the generation and evaluation of another candidate feature subset. The process of feature selection is illustrated in Fig. 1.

As mentioned in Sect. 1, feature selection techniques can be specialized into two categories, namely, filter and wrapper. The main difference between the two types of feature selection is in terms of the way of feature evaluation. The filter approach employs heuristics to rank the features according to their importance, whereas the wrapper approach employs an algorithm to learn classifiers from different subsets of features and then check the performance of these classifiers for evaluating the corresponding feature subsets. In terms of evaluation functions, popular heuristics employed by the filter approach include distance functions (Montalto et al. 2012), entropy (Shannon 1948), information gain (Kullback and Leibler 1951), correlation coefficients (Yu and Liu 2003), and co-variance (Barber 2012). The wrapper approach just simply employs the error rate of a classifier as the evaluation function (Dash and Liu 1997).

In terms of the performance of feature selection, the filter approach involves evaluation of features regardless of the fitness of the employed learning algorithm. In other words, a set of features are evaluated and the relevant ones are selected without considering that the selected feature subset is suitable or not for the chosen algorithm to learn a classifier. According to the experimental results reported in Dash and Liu (1997), feature selection through the filter approach leads to a low level of time complexity. However, when the selected feature subset is used for a pre-employed algorithm to learn a classifier, the error rate of classification may be high due to the case that the feature subset is not suitable for the algorithm to undertake learning tasks (Guyon 2003).

In contrast, the wrapper approach involves evaluation of features through checking the accuracy of the classifiers learned from different subsets of features. In other

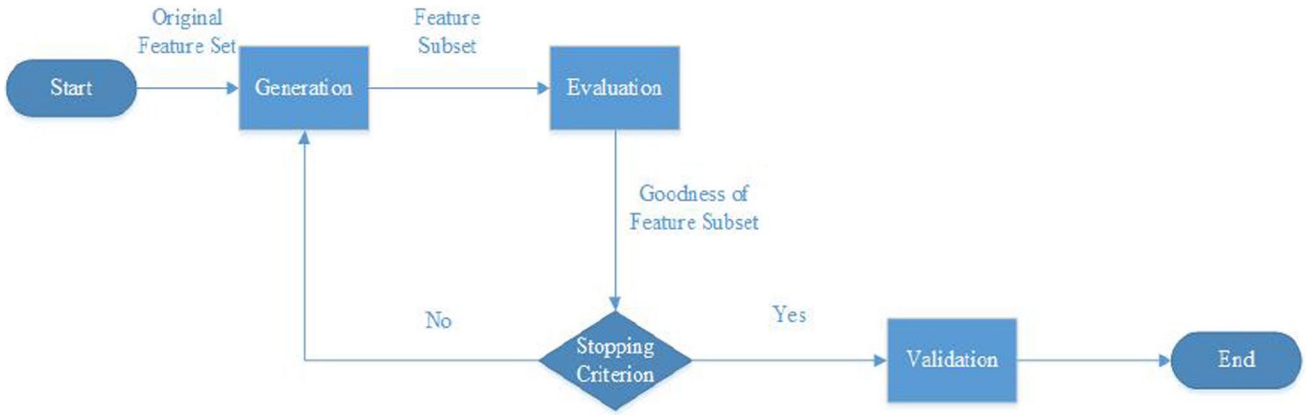

Fig. 1 Feature section process (Liu et al. 2017a)

words, a number (n) of different feature subsets are provided and an algorithm is used to learn n classifiers from these feature subsets. The feature subset, which leads to the production of the best classifier, is selected. According to the experimental results reported in Dash and Liu (1997), feature selection through the wrapper approach leads to very high accuracy of classification, but the time complexity is very high, due to the case that all the possible combinations of features (leading to different feature subsets) need to be examined.

Moreover, as argued in Cendrowska (1987), a feature may be only relevant to one class but irrelevant to all the other classes. Also, as mentioned in Sect. 1, in some application areas such as image processing, features are typically extracted from a specific target region. In this context, features could be only relevant to the class corresponding to the target region from which the features are extracted. From this point of view, if features extracted from different target regions are put together to make up a feature set, then it would result in a sparse matrix present in the feature set. In this case, it would be very likely to occur that some features are highly important for a specific class but are removed from the feature set due to low occurrence (and to reduce sparcity).

On the basis of the above argumentation, it is necessary to incorporate class-specific feature selection into both the filter and wrapper approaches. In particular, we consider traditional feature selection approaches (filter and wrapper) to be of single task and a framework of multi-task feature selection will be presented in Sect. 3.2.

3 Multi-task learning framework

In this section, we present a framework of multi-task learning. In particular, we describe how fuzzy approaches can be used to achieve multi-task classification and justify the significance of this way of classification. Also, we describe how Prism (a rule learning algorithm) can be used to achieve multi-task feature selection, i.e., feature selection for each class, and justify the significance of this way of feature selection.

The multi-task learning framework is illustrated in Fig. 2. In particular, a subset of features is selected for each specific class in the feature selection stage, i.e., all the features in a subset are highly relevant to a specific class (referred to as target class). In the training stage, from each feature set, a classifier is learned towards identifying instances of a target class, i.e., each classifier corresponds to a target class. In the classification stage, each classifier is used to identify whether an instance belongs to the target class (corresponding to this classifier). Finally, the outputs from these classifiers may need to be aggregated towards having a unique output, depending on the nature of the classification task, e.g., recognition, rating and decision making.

In general, multi-task feature selection is undertaken towards serving multi-task classification. However, multi-task classification can also be done independently, without the need to take multi-task feature selection. In Sect. 3.1, we will present how to achieve multi-task classification without feature selection. In Sect. 3.2, we will present how to achieve multi-task feature selection towards advances in classification performance.

3.1 Fuzzy multi-task classification

Multi-task classification is generally aimed at judging the membership or non-membership of an instance independently for each class. In particular, we adopt fuzzy approaches to measure the membership degree of an instance to each class.

Fuzzy classification is based on fuzzy logic, which is an extension of deterministic logic, i.e., the truth values (in

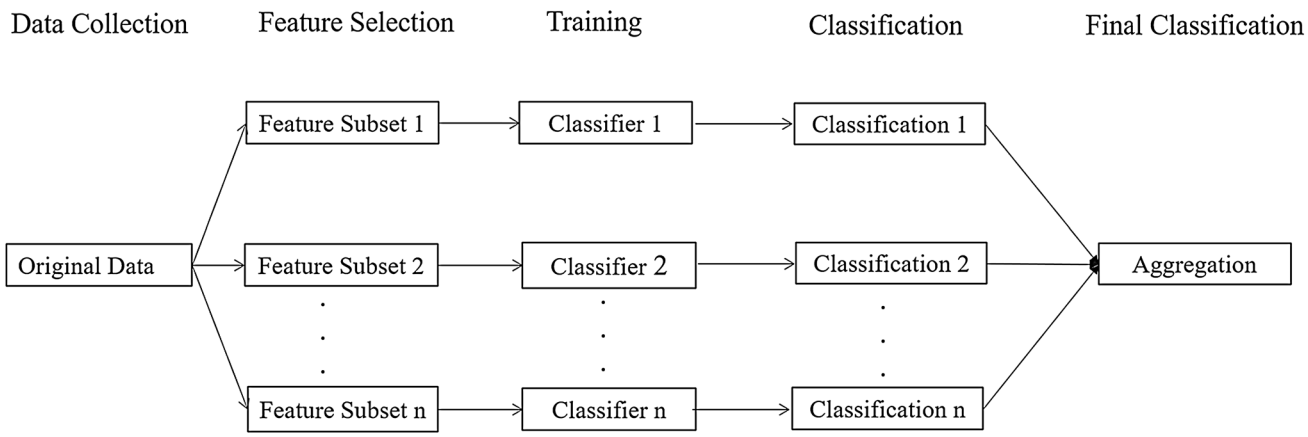

Fig. 2 Multi-task learning framework

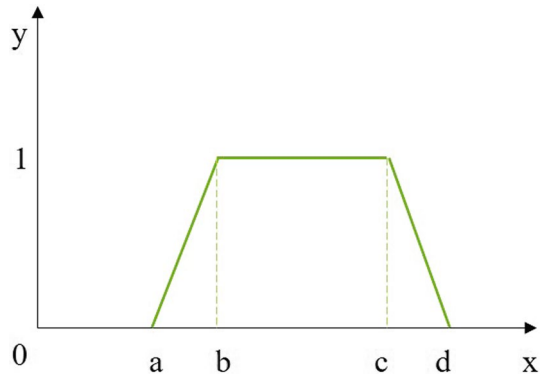

Fig. 3 Trapezoid membership function (Liu and Cocea 2017b)

the context of fuzzy logic) are ranged from 0 to 1 rather than binary (0 or 1). Fuzzy logic is typically used in the forms of fuzzy sets and fuzzy rule-based systems.

In the context of fuzzy sets, each element x_i belongs to the set S to a certain degree of membership. The value of the membership degree is dependent on the membership function $f_s(x_i)$ defined for the fuzzy set S. Membership functions are of various shapes, such as trapezoid, triangle and rectangle. In general, trapezoidal membership functions can be seen as a generalization of triangular and rectangular membership functions. A membership function is essentially defined by estimating four parameters a, b, c, d, as illustrated below and in Fig. 3.

$$f_T(x) = \begin{cases} 0, & \text{when } x \le a \text{ or } x \ge d; \\ (x-a)/(b-a), & \text{when } a < x < b; \\ 1, & \text{when } b \le x \le c; \\ (d-x)/(d-c), & \text{when } c < x < d; \end{cases}$$

According to Fig. 3, the shape of the membership function would be triangle, if $b = c$, or the shape would be rectangle, if $a = b$ and $c = d$. A membership function can be defined using expert knowledge (Mamdani and Assilian 1999) or by learning statistically from data (Bergadano and Cutello 1993). More details on fuzzy sets and logic can be found in (Zadeh 1965; Chen and Chang 2001; Chen and Chen 2011; Chen 1996).

In the context of fuzzy rule based systems, the main operations include fuzzification of continuous attributes and learning of fuzzy rules.

In terms of fuzzifying continuous attributes, it is needed to determine the number of linguistic terms to be transformed from a continuous attribute. Furthermore, a membership function needs to be defined for each of the linguistic terms, i.e., a linguistic term is viewed as a fuzzy set and the domain of the term is [0, 1], so a membership function needs to be defined for mapping a value of a continuous attribute into a membership degree value of a linguistic term (transformed from the continuous attribute).

Following the fuzzification of continuous attributes, a number of rules can be learned from data and the best rules can be used for predictions on new data. Some methods of fuzzy rules learning can be found in Wang and Mendel (1992), Chen and Lee (2010), and Berthold (2003). In the context of fuzzy rule based classification, following the learning stage, the resulting rules are typically represented in the following form:

- Rule 1: if x_1 is A_{11} and x_2 is A_{21} and ...and x_n is A_{n1} then class= C_1;
- Rule 2: if x_1 is A_{12} and x_2 is A_{22} and ...and x_n is A_{n2} then class = C_2;
- ⋮
- ⋮
- Rule m: if x_1 is A_{1m} and x_2 is A_{2m} and ...and x_n is A_{nm} then class = C_k;

A_{nm} represents a linguistic term, where n is the index of attribute A and m is the index of rule. Also, C_k represents a class label, where k is the class index.

In the context of multi-task classification, a fuzzy rule based system is used following the four steps: fuzzification, application, implication and aggregation. We illustrate the whole procedure by using the following example of fuzzy rules:

- Rule 1: if x_1 is Young and x_2 is Long then class = Positive;
- Rule 2: if x_1 is Young and x_2 is Middle then class = Neutral;
- Rule 3: if x_1 is Young and x_2 is Short then class = Negative;
- Rule 4: if x_1 is Middle-aged and x_2 is Long then class = Neutral;
- Rule 5: if x_1 is Middle-aged and x_2 is Middle then class = Positive;
- Rule 6: if x_1 is Middle-aged and x_2 is Short then class = Negative;
- Rule 7: if x_1 is Old and x_2 is Long then class = Negative;
- Rule 8: if x_1 is Old and x_2 is Middle then class = Positive;
- Rule 9: if x_1 is Old and x_2 is Short then class = Neutral;

The fuzzy membership functions defined for the linguistic terms transformed from x_1 and x_2 are illustrated in Figs. 4 and 5, respectively.

According to Figs. 4 and 5, if $x_1 = 30$ and $x_2 = 47$, then the following steps will be executed:

Fuzzification:

Rule 1: $f_{Young}(30) = 0.67, f_{Long}(47) = 0.7;$
Rule 2: $f_{Young}(30) = 0.67, f_{Middle}(47) = 0.3;$
Rule 3: $f_{Young}(30) = 0.67, f_{Short}(47) = 0;$
Rule 4: $f_{Middle-aged}(30) = 0.33, f_{Long}(47) = 0.7;$
Rule 5: $f_{Middle-aged}(30) = 0.33, f_{Middle}(47) = 0.3;$
Rule 6: $f_{Middle-aged}(30) = 0.33, f_{Short}(47) = 0;$
Rule 7: $f_{Old}(30) = 0, f_{Long}(47) = 0.7;$
Rule 8: $f_{Old}(30) = 0, f_{Middle}(47) = 0.3;$

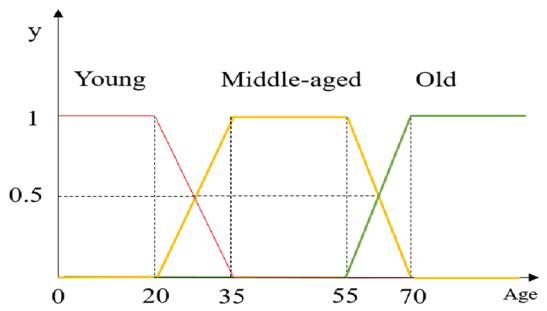

Fig. 4 Membership functions for linguistic terms of attribute 'age'

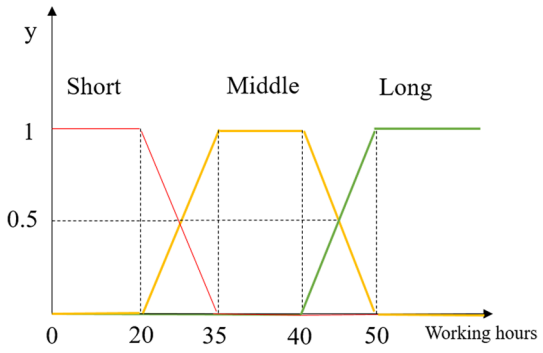

Fig. 5 Membership functions for linguistic terms of attribute 'working hours'

Rule 9: $f_{Old}(30) = 0, f_{Short}(47) = 0;$

In the fuzzification step, the notation $f_{Long}(47) = 0.7$ represents that the membership degree of the numerical value '47' to the fuzzy set defined with the linguistic term 'Long' is 0.7. The fuzzification step is aimed at mapping the value of a continuous attribute to a value of membership degree to a fuzzy set (i.e., mapping to the value of a linguistic term transformed from the continuous attribute).

Application:

Rule 1: $f_{Young}(30) \wedge f_{Long}(47) = Min(0.67, 0.7) = 0.67;$
Rule 2: $f_{Young}(30) \wedge f_{Middle}(47) = Min(0.67, 0.3) = 0.3;$
Rule 3: $f_{Young}(30) \wedge f_{Short}(47) = Min(0.67, 0) = 0;$
Rule 4: $f_{Middle-aged}(30) \wedge f_{Long}(47) = Min(0.33, 0.7) = 0.33;$
Rule 5: $f_{Middle-aged}(30) \wedge f_{Middle}(47) = Min(0.33, 0.3) = 0.3;$
Rule 6: $f_{Middle-aged}(30) \wedge f_{Short}(47) = Min(0.33, 0) = 0;$
Rule 7: $f_{Old}(30) \wedge f_{Long}(47) = Min(0, 0.7) = 0;$
Rule 8: $f_{Old}(30) \wedge f_{Middle}(47) = Min(0, 0.3) = 0;$
Rule 9: $f_{Old}(30) \wedge f_{Short}(47) = Min(0, 0) = 0$

In the application step, the conjunction of the two fuzzy membership degrees, respectively, for the two attributes 'x_1 and 'x_2' is aimed at deriving the firing strength of a fuzzy rule. For example, the antecedent of Rule 1 consists of x_1 is Young and x_2 is Long, so the firing strength of Rule 1 is 0.67, while $f_{Young}(30) = 0.67$ and $f_{Long}(47) = 0.7$.

Implication:

Rule 1: $f_{Rule1 \rightarrow Positive}(30, 47) = 0.67;$
Rule 2: $f_{Rule2 \rightarrow Neutral}(30, 47) = 0.3;$
Rule 3: $f_{Rule3 \rightarrow Negative}(30, 47) = 0;$
Rule 4: $f_{Rule4 \rightarrow Neutral}(30, 47) = 0.33;$
Rule 5: $f_{Rule5 \rightarrow Positive}(30, 47) = 0.3;$
Rule 6: $f_{Rule6 \rightarrow Negative}(30, 47) = 0;$
Rule 7: $f_{Rule7 \rightarrow Negative}(30, 47) = 0;$
Rule 8: $f_{Rule8 \rightarrow Positive}(30, 47) = 0;$
Rule 9: $f_{Rule9 \rightarrow Neutral}(30, 47) = 0;$

In the implication step, the firing strength of a fuzzy rule derived in the application step can be used further to identify the membership degree of the value of an input vector to the class label 'Positive', 'Neutral' or 'Negative', depending on the consequent of the fuzzy rule. For example, $f_{Rule1 \rightarrow Positive}(30, 47) = 0.67$ indicates that the consequent of Rule 1 is assigned the class label 'Positive' and the input vector '(30, 47)' has the membership degree of 0.67 to the class label 'Positive'. In other words, the inference through Rule 1 leads to the input vector '(30, 47)' having the membership degree value of 0.67 to the class label 'Positive'.

Aggregation:

$$f_{Positive}(30,47) = f_{Rule1 \rightarrow Positive}(30,47) \vee f_{Rule5 \rightarrow Positive}(30,47) \vee f_{Rule8 \rightarrow Positive}(30,47) = Max(0.67, 0.3, 0) = 0.67$$

$$f_{Neutral}(30,47) = f_{Rule2 \rightarrow Neutral}(30,47) \vee f_{Rule4 \rightarrow Neutral}(30,47) \vee f_{Rule9 \rightarrow Neutral}(30,47) = Max(0.3, 0.33, 0) = 0.33$$

$$f_{Negative}(30,47) = f_{Rule3 \rightarrow Negative}(30,47) \vee f_{Rule6 \rightarrow Negative}(30,47) \vee f_{Rule7 \rightarrow Negative}(30,47) = Max(0, 0, 0) = 0$$

In the aggregation step, the membership degree value of the input vector to the class label ('Positive', 'Neutral' or 'Negative'), which is inferred through a rule, is compared with the other membership degree values inferred through the other rules, towards finding the maximum among all the membership degree values. For example, Rule 1, Rule 5 and Rule 8 are all assigned the class label 'Positive' as their consequent and the membership degree values of the input vector '(30, 47)' inferred through the three rules are 0.67, 0.3 and 0, respectively, to the class label 'Positive'. As the maximum of the fuzzy membership degree values is 0.67, the input vector is considered to have the membership degree value of 0.67 to the class label 'Positive'.

In traditional machine learning, the classification outcome needs to be crisp so defuzzification is typically involved by choosing the class label with the highest value of membership degree. For the above example, the final classification outcome is to assign the class label 'Positive' to the unseen instance '(30, 47, ?)', since the value (0.67) of the membership degree to this class label is the highest. In contrast, as mentioned above, multi-task classification is aimed at measuring the membership degree value of an instance to each class, so it is not necessary to use defuzzification.

On the other hand, multi-task classification can be done for providing either a single crisp output (a unique class label) or multiple fuzzy outputs (membership degree values for these class labels), depending on the nature of the classification task. In particular, the former way of classification is typically taken only when different classes are assumed to be mutually exclusive (like the above illustrative example). In this context, the outcome of fuzzy classification would typically show that the sum of the membership degree values of an instance to the given class labels is 1. The above outcome is mainly due to the case that the linguistic terms transformed from the same continuous attribute are considered to be mutually exclusive. For example, both Figs. 4 and 5 show that for the same horizontal coordinate value (the value of a continuous attribute) the sum of the corresponding vertical coordinate values (membership degree values) is always 1, due to the constraint that the linguistic terms (e.g., 'Long', 'Middle' and 'Short') are defined to be mutually exclusive.

However, as argued in Sect. 1, there are many real-life examples indicating that different classes are not mutually exclusive. The argumentation can also be supported in the context of fuzzy classification, as illustrated in Fig. 6. In particular, the membership functions, which are learned from the Anneal dataset regarding the carbon attribute, can show that the sum of the membership degree values could be higher than 1. In this case, the outcome of fuzzy classification would show that an instance belongs to more than one class with a high value of membership degree (close or even equal to 1), as reported in Liu et al. (2017b).

In addition, the outcome of generative multi-task classification can also show that an instance has the fuzzy membership degree value of 0 to all the given class labels. The above phenomenon can be explained by the possible case that the set of given class labels is not complete and it is needed to add an extra class label to which the instance actually belongs.

From mathematical perspective, the above explanation can be supported by the concept that a function f is defined as a mapping from set A to set B and the range R of this function f is a subset of set B. From this point of view, if a function f is not a complete mapping, then not all elements of set S are in the range R of this function. A classifier is essentially a function that provides a discrete value as the output, so it is possible that an instance cannot be classified due to the case of an incomplete mapping.

On the basis of the above argumentation, it is necessary to turn discriminative single-task classification into generative multi-task classification. We will show experimental results in Sect. 4 to support the argumentation.

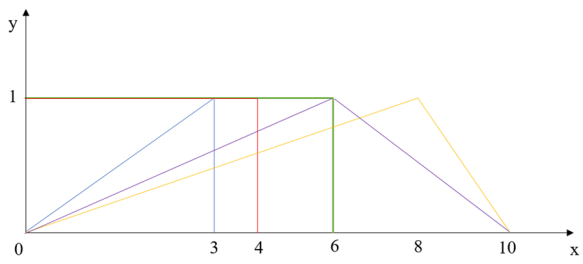

Fig. 6 Membership functions learned from anneal data on carbon attribute (Liu and Cocea 2018)

3.2 Multi-task feature selection

Multi-task feature selection generally means to select a subset of features for each class, since a feature may not be relevant to all the classes, as argued in Sect. 1. In particular, we propose to use the Prism algorithm (Cendrowska 1987) towards class-specific feature selection.

Prism is a rule learning algorithm that follows the separate and conquer strategy (Furnkranz 1999). The algorithm is capable of self-evaluation of features in terms of their relevance to a specific class. The procedure of this algorithm is shown in Algorithm 1.

Algorithm 1: Prism Algorithm (Liu *et al.*, 2016a)

Input : a training set T, a subset $T' \subseteq T$, an attribute set AS, an instance $t \in T$, dimensionality d, an attribute A_x (x is the index of the attribute), class C_i (i is the index of the class), number of classes n

Output: a rule set RS, a result set of instances T" covered by a rule $R \in RS$

```
Initialize: T' = T, T" = T', i = 0;
for i < n do
    while ∃t : t ∈ T' ∧ t ∈ C_i do
        generate a rule for class C_i
        while ∃t : t ∈ T' ∧ t ∉ C_i do
            x = 0;
            while x < d do
                for each value v of A_x do
                    Calculate P(C_i|A_x = v);
                end
                x++;
            end
            assign A_x = v to R as a rule term, when
            P(C_i|A_x = v) is max;
            AS= AS - {A_x};
            d= d − 1
            ∀t : T" = T" − {t}, if t ∈ T' and t does
            not comprise A_x = v;
        end
        RS= RS ∪ {R};
        T'= T' - T";
    end
    T' = T;
    i++;
end
```

It can be seen from Algorithm 1 that the Prism algorithm needs to select a class as the target class towards learning a set of rules that discriminate the target class from all the other classes. In particular, each of the n classes is selected in turn as the target class, so there are n sets of rules learned from the same training set, i.e., the learning of each set of rules of a specific class is separate from the learning of other sets of rules of other classes.

We use the contact lenses data set (Cendrowska 1987) as an example for illustrating the Prism algorithm. The details of the dataset are shown in Table 5.

In this data set, there are three classes, namely, 'no lenses', 'soft lenses' and 'hard lenses', so there would be three sets of rules learned for the three classes, respectively.

According to Table 5, we can get a frequency table for each attribute, i.e., we have four frequency tables for the four attributes: 'age' (see Table 6, 'spectacle-prescrip' (see Table 7), 'astigmatism' (see Table 8) and 'tear-prod-rate' (see Table 9).

Based on the frequency tables, the conditional probabilities for each attribute–value pair of each attribute can be calculated. We display these here for ease of explanation—in the normal course of the algorithm, the probabilities would be calculated when needed, not in advance.

According to Table 6, we can derive the conditional probability for each of the three values of attribute 'age', towards each of the three classes.

$$P(class = hard\ lenses|age = young) = \frac{2}{8}$$
$$P(class = hard\ lenses|age = pre\text{-}presbyopic) = \frac{1}{8}$$
$$P(class = hard\ lenses|age = presbyopic) = \frac{1}{8}$$
$$P(class = soft\ lenses|age = young) = \frac{2}{8}$$
$$P(class = soft\ lenses|age = pre\text{-}presbyopic) = \frac{2}{8}$$
$$P(class = soft\ lenses|age = presbyopic) = \frac{1}{8}$$
$$P(class = no\ lenses|age = young) = \frac{4}{8}$$
$$P(class = no\ lenses|age = pre\text{-}presbyopic) = \frac{5}{8}$$
$$P(class = no\ lenses|age = presbyopic) = \frac{6}{8}$$

According to Table 7, we can derive the conditional probability for each of the two values of attribute 'spectacle-prescrip', towards each of the three classes.

$$P(class = hard\ lenses|\ spectacle\text{-}prescrip = myope) = \frac{3}{12}$$
$$P(class = hard\ lenses|spectacle\text{-}prescrip = hypermetrope) = \frac{1}{12}$$
$$P(class = soft\ lenses|spectacle\text{-}prescrip = myope) = \frac{2}{12}$$
$$P(class = soft\ lenses|spectacle\text{-}prescrip = hypermetrope) = \frac{3}{12}$$
$$P(class = no\ lenses|spectacle\text{-}prescrip = myope) = \frac{7}{12}$$
$$P(class = no\ lenses|\ spectacle\text{-}prescrip = hypermetrope) = \frac{8}{12}$$

According to Table 8, we can derive the conditional probability for each of the two values of attribute 'astigmatism', towards each of the three classes.

$$P(class = hard\ lenses|astigmatism = no) = \frac{0}{12}$$
$$P(class = hard\ lenses|astigmatism = yes) = \frac{4}{12}$$
$$P(class = soft\ lenses|astigmatism = no) = \frac{5}{12}$$
$$P(class = soft\ lenses|astigmatism = yes) = \frac{0}{12}$$
$$P(class = no\ lenses|astigmatism = no) = \frac{7}{12}$$
$$P(class = no\ lenses|astigmatism = yes) = \frac{8}{12}$$

Table 5 Contact lenses data

Age	Spectacle-prescrip	Astigmatism	Tear-prod-rate	Class
Young	Myope	No	Reduced	No lenses
Young	Myope	No	Normal	Soft lenses
Young	Myope	Yes	Reduced	No lenses
Young	Myope	Yes	Normal	Hard lenses
Young	hypermetrope	No	Reduced	No lenses
Young	Hypermetrope	No	Normal	Soft lenses
Young	Hypermetrope	Yes	Reduced	No lenses
Young	Hypermetrope	Yes	Normal	Hard lenses
Pre-presbyopic	Myope	No	Reduced	No lenses
Pre-presbyopic	Myope	No	Normal	Soft lenses
Pre-presbyopic	Myope	Yes	Reduced	No lenses
Pre-presbyopic	Myope	Yes	Normal	Hard lenses
Pre-presbyopic	Hypermetrope	No	Reduced	No lenses
Pre-presbyopic	Hypermetrope	No	Normal	Soft lenses
Pre-presbyopic	Hypermetrope	Yes	Reduced	No lenses
Pre-presbyopic	Hypermetrope	Yes	Normal	No lenses
Presbyopic	Myope	No	Reduced	No lenses
Presbyopic	Myope	No	Normal	No lenses
Presbyopic	Myope	yes	reduced	No lenses
Presbyopic	Myope	Yes	Normal	Hard lenses
Presbyopic	Hypermetrope	No	Reduced	No lenses
Presbyopic	Hypermetrope	No	Normal	Soft lenses
Presbyopic	Hypermetrope	Yes	Reduced	No lenses
Presbyopic	Hypermetrope	Yes	Normal	No lenses

Table 6 Frequency table for age

Class label	Age = young	Age = pre-presbyopic	Age = presbyopic
Hard lenses	2	1	1
Soft lenses	2	2	1
No lenses	4	5	6
Total	8	8	8

Table 8 Frequency table for astigmatism

Class label	Astigmatism = no	Astigmatism = yes
Hard lenses	0	4
Soft lenses	5	0
No lenses	7	8
Total	12	12

Table 7 Frequency table for spectacle-prescrip

Class label	Spectacle-prescrip = myope	Spectacle-prescrip = hypermetrope
Hard lenses	3	1
Soft lenses	2	3
No lenses	7	8
Total	12	12

Table 9 Frequency table for tear-prod-rate

Class label	Tear-prod-rate = reduced	Tear-prod-rate = normal
Hard lenses	0	4
Soft lenses	0	5
No lenses	12	3
Total	12	12

According to Table 9, we can derive the conditional probability for each of the two values of attribute 'tear-prod-rate', towards each of the three classes.

$$P(class = hard\ lenses|tear\text{-}pro\text{-}rate = reduced) = \frac{0}{12}$$
$$P(class = hard\ lenses|tear\text{-}prod\text{-}rate = normal) = \frac{4}{12}$$
$$P(class = soft\ lenses|tear\text{-}prod\text{-}rate = reduced) = \frac{0}{12}$$

$$P(class = soft\ lenses|tear\text{-}prod\text{-}rate = normal) = \frac{5}{12}$$
$$P(class = no\ lenses|tear\text{-}prod\text{-}rate = reduced) = \frac{12}{12}$$
$$P(class = no\ lenses|tear\text{-}prod\text{-}rate = normal) = \frac{3}{12}$$

When the target class is 'no lenses', the first attribute, i.e., 'tear-prod-rate', is selected (line 6 in Algorithm 1) and the attribute–value pair (*tear-prod-rate = reduced* or *tear-prod-rate = normal*) with the maximum conditional probability is chosen (line 13 in Algorithm 1). Of the two attribute-value pairs, *tear-prod-rate = reduced* has the maximum conditional probability, i.e., $P(class = no\ lenses|tear\text{-}prod\text{-}rate = reduced) = 1$.

Since the maximum probability is reached, i.e. 1, the learning of the first rule is complete and the first rule learned is expressed as: if *tear-prod-rate = reduced* then *class = no lenses*. Following the completion of learning the first rule, all the 12 instances with the attribute-value pair *tear-prod-rate = reduced* are deleted from the training set, and the learning of the second rule is started on the reduced training set.

The above illustration indicates that the nature of the Prism algorithm is to evaluate each attribute–value pair in terms of their importance to a specific class. For example, the probability $P(class = no\ lenses|tear\text{-}prod\text{-}rate = reduced) = 1$ indicates that the attribute value pair *tear-prod-rate = reduced* is highly important for the 'no lenses' class. In other words, the attribute value pair *tear-prod-rate = reduced* is selected as the only term of the rule: if *tear-prod-rate = reduced* then *class = no lenses*, which indicates that the attribute 'tear-prod-rate' is relevant to the class 'no lenses'. In contrast, if an attribute has never appeared (alongside one of its values) as a rule term in any rules of a specific class, then it would indicate empirically that the attribute is not relevant to the class.

On the basis of the above illustration and argumentation, the Prism algorithm is judged to be capable of self-evaluation of features (attributes) in terms of their relevance to a specific class. We will show experimental results in Sect. 4 to this effect.

4 Experiments, results and discussion

In this section, we report two experimental studies for multi-task classification and feature selection, respectively. In particular, we use the fuzzy rule learning approach for multi-task classification to show that an instance may belong to more than one class or may not belong to any one of the classes. We also compare the fuzzy approach with three popular discriminative learning approaches (C4.5, Naive Bayes and K Nearest Neighbour), in terms of the classification performance on instances that may belong to more than one class. In terms of multi-task feature selection, we use the Prism algorithm for evaluating features in terms of their relevance to each specific class.

Table 10 Characteristics of data sets

Dataset	Attribute types	Attributes	Instances	Classes
ERA	Continuous	4	1000	9
ESL	Continuous	4	488	9
LEV	Continuous	4	1000	5
SWD	Continuous	10	1000	4

For the experimental study on multi-task classification, we use four real-world data sets retrieved from the Weka distribution (David 2005). The characteristics of the four data sets are shown in Table 10. In particular, the ERA dataset contains information on job applications and the output is the degree to which an applicant is acceptable or not. The ESL contains information on job applications as well but the output is the degree to which an applicant is suitable to a specific type of job. The LEV dataset contains information on teaching assessment and the output is an overall evaluation of a lecturer's performance. The SWD dataset contains information on real-world assessment of qualified social workers and the output is the degree of risk facing children if they stay with their families at home. For all of the four data sets, the output is actually ordinal, which indicates that the classification tasks are for the purpose of rating, as described in Sect. 2.2.

The results of fuzzy rule based classification on the four data sets are shown in Tables 11, 12, 13 and 14, in terms of the membership degree values of the instances to the class labels (selected as the representative examples). The first column, i.e., out1, represents the true class label, while the last column, i.e., prediction, represents the output from the fuzzy classification. The results show that it is possible to have an instance belong to more than one class, i.e., the instance has a very high value of membership degree (close or even equal to 1) to more than one class.

As mentioned above, all of the four data sets contain information about subjective evaluation, i.e., the classification tasks involved in the four data sets belong to rating, since the outputs are all ordinal. In this context, the phenomenon that an instance may belong to more one class could be explained by the nature case that the same item can be given very different ratings by different users, since they may have different backgrounds and preferences.

From machine learning perspective, an instance is labelled subjectively by a person randomly selected, but people who have different backgrounds and preferences from this person would be very likely to provide the same instance with other labels. Moreover, a training set could contain instances which are highly similar to each other but are provided with different labels, due to the case that these

Table 11 Results sample on ERA data

out1	1	2	3	4	5	6	7	8	9	Prediction
5	1	1	1	1	1	1	0	0	0	5
4	0	0	1	1	1	1	1	1	1	4
8	0	0	1	0	1	1	1	1	0	8
8	1	0	0	0	0	1	1	1	1	8
5	1	1	1	1	1	1	0	0	0	5
9	0	0	1	0	0	1	1	1	1	9
9	0	0	1	0	0	1	1	1	1	9
4	1	1	1	1	1	1	0	0	0	4
6	1	1	1	1	1	1	1	1	1	6
3	0	0	1	1	1	1	1	0	0	3

Table 12 Results sample on ESL data

out1	1	2	3	4	5	6	7	8	9	Prediction
4	0	0	1	1	0	0	0	0	0	4
7	0	0	0	0	0	1	1	1	0	7
5	0	0	0	0	1	0	0	0	0	5
6	0	0	0	0	0	1	0	0	0	6
3	0	0	1	0	0	0	0	0	0	3
2	0	1	0	0	0	0	0	0	0	2
5	0	0	0.5	0	1	0	0	0	0	5
6	0	0	0	0	0	0	0	0	0	?
3	0	0	0	0	0	0	0	0	0	?
2	0.75	1	0	0.5	0	0	0	0	0	2

Table 13 Results sample on LEV data

out1	0	1	2	3	4	Prediction
2	0	1	1	0	0	2
2	0	1	1	0	0	2
1	0	1	1	0	0	1
3	0	0	0	1	1	3
2	0	1	1	1	0	2
3	0	0	1	1	1	3
4	0	0	0	1	1	4
3	0	0	0	1	0	3
2	0	0	1	0	0	2
0	1	0	0	0	0	0

Table 14 Results sample on SWD data

out1	2	3	4	5	Prediction
4	0	1	1	0	4
4	1	1	1	0	4
3	0	1	1	1	3
5	0	0	1	1	5
3	0	1	1	0	3
2	1	1	1	0	2
2	1	1	0	0	2
3	0	1	0	0	3
4	0	0	1	0	4
5	0	0	0	1	5

instances are labelled by people who are highly dissimilar to each other, in terms of their backgrounds and preferences.

In common sense, subjective evaluation generally means that people are biased on some particular aspects. For example, a soldier may be very good at fighting and has got a lot of military awards, but the solider may also have made a lot of mistakes in daily life, leading to disciplinary actions being taken against the solider. In this case, it is very difficult to say that this is a good or bad solider. For military commanders, the capability of fighting may be treated as more important, so they may be biased towards saying that this is a good solider. In contrast, for political officers, the behavior of a soldier in daily life may be considered to have a higher impact, so they may be biased towards saying that this is a bad solider.

In the context of machine learning, the above example could indicate that people may do data labelling without considering all the provided features, i.e, they may provide an instance with a label based only on those features that they think of higher importance. In the context of generative multi-task classification through fuzzy approaches, the above example can indicate that the solider could have a

very high membership degree to both the 'good' and 'bad' classes, due to subjective evaluations from different kinds of people.

The above argumentation can also be supported by the results shown in Tables 11, 12, 13 and 14. For example, the first two data sets are about evaluation of the acceptance degree and the suitability of each job applicant. In this context, each applicant may have strengths and weaknesses reflected from the values of different features, so different recruiters may have different opinions on the acceptance degree and the suitability of an applicant to a particular job, unless the applicant is extremely strong in all these criteria or does not meet these criteria. From this point of view, it is generally likely that a job applicant has a high membership degree to more than one class. The same argumentation also applies to the results on the other two data sets. For example, the third dataset is about evaluation of teaching performance of lecturers, and judgment could usually be subjective, due to different opinions from different people on what constitute professional teaching.

In real applications, the case that a fuzzy classifier provides an instance with multiple highest values of membership degree would be considered as the possibility of having different labels assigned to the same instance by different kinds of people.

In addition, the results shown in Table 12 can indicate that it is possible that an instance has the membership degree value of 0 to all the classes. This phenomenon generally indicates that the instance does not belong to any one of the given classes and thus an extra class needs to be provided, as mentioned in Sect. 3.1. However, in the context of rating, the above phenomenon is more likely due to the case that the profile of a particular applicant is a very special example and does not have any similarity to all of the other applicants'. In other words, the applicant has a profile much stronger or weaker than all of the other applicants'. In the context of fuzzy rule based classification, this indicates that for each class none of the rules fires, i.e., the firing strength of each of these rules is 0. This also indicates that fuzzy classifiers generally have no bias towards one class and against all the other classes, unlike classifiers produced by discriminative learning approaches.

In terms of classification accuracy (obtained using cross validation), the fuzzy rule learning approach is compared with C4.5, Naive Bayes (NB), K Nearest Neighbour (KNN). The results are shown in Table 15. In particular, the results show that the fuzzy approach outperforms significantly the three discriminative learning approaches in all of the four cases. The results are very likely due to the case that the fuzzy approach aims at classifying each instance through measuring independently the membership degree value of an instance to each class, unlike the discriminative learning approaches, which aim at discriminating one class from the other classes towards assigning an instance a unique class. In other words, in the context of generative multi-task classification, the fuzzy approach leads to correct classification when identifying the case that the membership degree value of the instance to the class (labelled as the ground truth) is the highest one (usually close or even equal to 1), without the need to discriminate this class from the other classes.

Table 15 Classification accuracy

Dataset	C4.5	NB	KNN	Fuzzy
ERA	0.264	0.262	0.274	0.934
ESL	0.656	0.662	0.676	0.805
LEV	0.621	0.557	0.631	0.925
SWD	0.570	0.582	0.568	0.916

For the experimental study on multi-task feature selection, the results are shown in Figs. 7 and 8. Also, there are 247 rules learned from the image segmentation dataset retrieved from the UCI repository (Lichman 2013). In addition, through ten-fold cross validation of the Prism algorithm learning from the above data set, the classification accuracy is 92%, which indicates that the rules learned by the algorithm are trust-worthy.

Figure 7 represents the frequency of each feature used in rules of each class and shows that some features are frequently used in general, such as 'region-centroid-row', 'intensity-mean' and 'hue-mean'. However, through looking at the frequencies of these features used for each specific class, it can be indicated that the features are more frequently used in rules of one or two classes but are much less frequently or even never used in rules of all the other classes. For example, the total frequency of the 'region-centroid-row' feature is 106, which means that the feature is used in 106 out of the all 247 rules. However, the frequency of the feature is very different for different classes. In particular, the frequency is 45 for cement, 28 for window, 21 for foliage, 6 for path, 4 for brickface, 2 for grass, and 0 for sky.

Figure 8 represents how often a feature has been used in rules of a specific class. For example, there are 15 rules of the 'brickface' class and the 'hue-mean' feature has been used in 11 out of the 15 rules, which indicates that the selection rate of the 'hue-mean' feature for the 'brickface' class is 0.73. The results on selection rate indicate that features selected for a specific class could be of different levels of relevance. For example, for the 'brickface' class, there are 10 out of the 19 features used in the 15 rules, namely, 'region-centroid-col' (used in 1 out of the 15 rules), 'region-centroid-row'(used in 4 out of the 15 rules), 'vedge-mean' (used in 1 out of the 15 rules), 'vegde-sd' (used in 1 out of the 15 rules), 'hedge-sd' (used in 1 out of the 15 rules), 'intensity-mean' (used in 3 out of the 15 rules), 'rawred-mean' (used in 2 out of the 15 rules), 'rawgreen-mean' (used in 4 out of

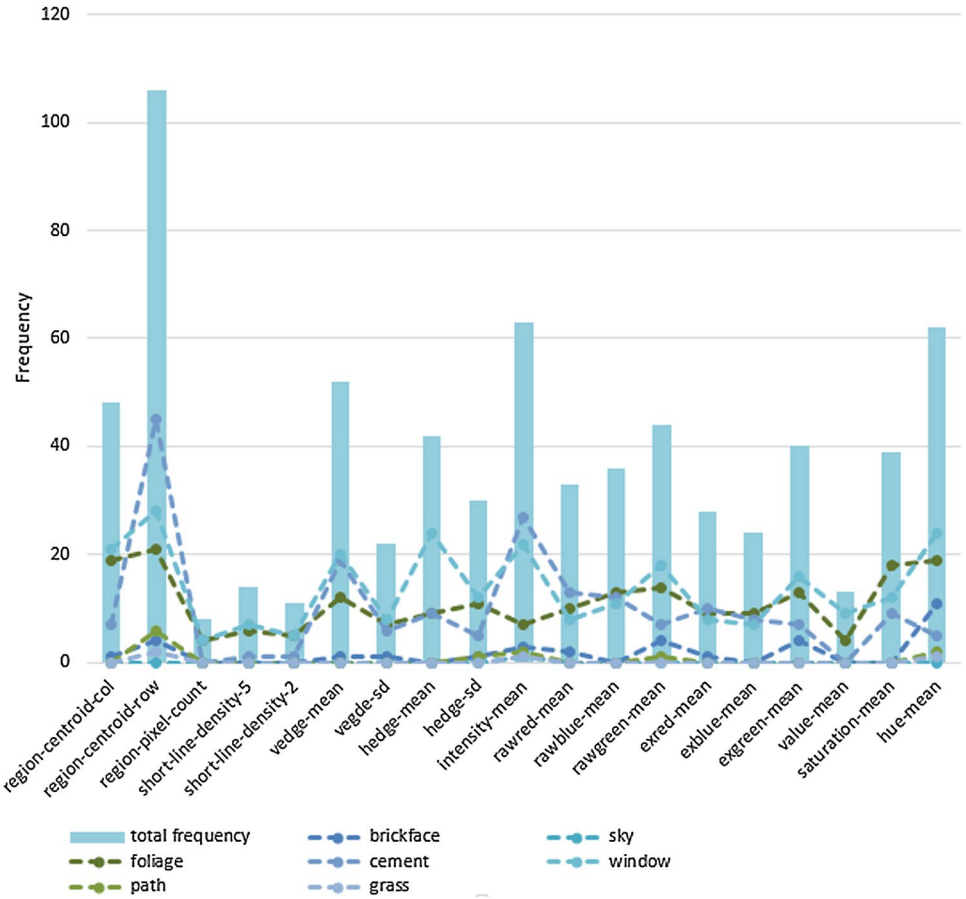

Fig. 7 Frequency of features selected in rules of a specific class

the 15 rules), 'exred-mean' (used in 1 out of the 15 rules), 'exgreen-mean' (used in 4 out of the 15 rules) and 'hue-mean' (used in 11 out of the 15 rules).

The results shown in Figs. 7 and 8 indicate that the Prism algorithm is capable of self-evaluation of features in terms of their relevance to each specific class. As mentioned above, rules of different classes learned by Prism include different features in the rule antecedents, i.e., for each class, only some of the features are selected for appending terms into the antecedents of the learned rules. The rules of different classes, which are used together as a rule based classifier, perform well on the image segmentation data set, as mentioned above.

In comparison with the traditional filter approach, the capacity of Prism in self-evaluation of features can lead to avoiding the case that some features are highly required for an algorithm to learn, but are removed from the feature set by a filtering-based feature selection method. In other words, a filtering-based method may judge independently some features as irrelevant without considering the fitness of these features to a particular learning algorithm.

In comparison with the traditional wrapper approach, use of the Prism algorithm for multi-task feature selection is likely to lead to the reduction of computational complexity, since the wrapper approach needs to evaluate all combinations of features and the dimensionality of image data is typically high. In other words, feature selection through the Prism algorithm or other rule learning algorithms is self-inclusive in the training stage, and all relevant features are already selected once the learning of a rule based classifier is complete. The selected features can also be used for other algorithms to learn classifiers.

5 Conclusions

In this paper, we proposed the framework of multi-task learning (a learning task per class) to deal with practical issues in classification and feature selection. In particular, we used fuzzy approaches for transforming discriminative single-task classification into generative multi-task classification. Also, we used the Prism algorithm for transforming single-task feature selection into multi-task feature selection, i.e., the Prism algorithm is used as a wrapper approach for evaluating features in terms of their relevance to each specific class and selecting a subset of features important for the class. The experimental results show that it is effective and leads to advances in prediction performance to adopt

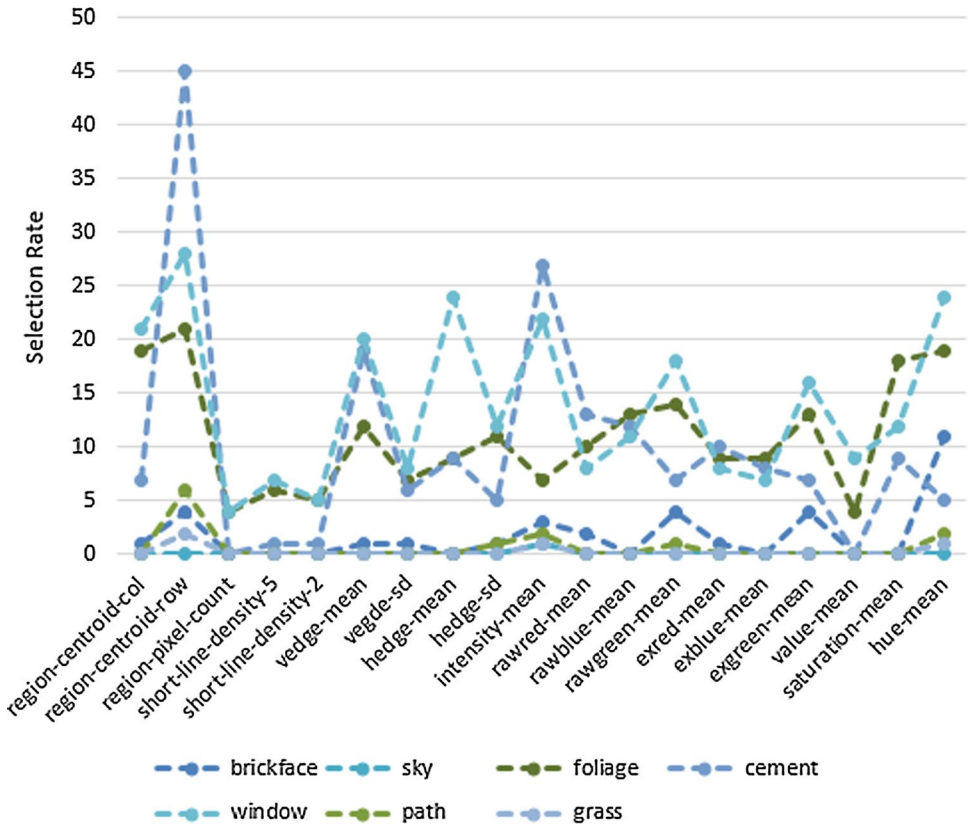

Fig. 8 Selection rate of features for a specific class

the proposed framework of multi-task learning for both classification and feature selection.

In terms of generative multi-task classification through fuzzy approaches, the experimental results show that an instance may have a high value of membership degree (close or even equal to 1) to more than one class. The results also show that an instance may have a membership degree value of 0 to all the classes, which indicates that the instance does not belong to any of the given classes and an extra class is thus needed to be added into the set of given classes. In comparison with popular algorithms that belong to discriminative learning, such as C4.5, Naive Bayes and K Nearest Neighbour, fuzzy approaches, which typically belong to generative learning, perform better accuracy of classification, when different classes may be correlated or mutually independent rather than mutually exclusive.

In terms of multi-task feature selection through using the Prism algorithm, seven sets of rules were learned from the image segmentation dataset and each set of rules is specific for one of the seven classes involved in the data set. Through checking each set of rules, it can be seen that some features are frequently used alongside their values as terms in different rules but some other features are never used in any rules. Also, it can be seen that the same feature is frequently used in the rules of one class, but is less frequently or even never used in the rules of the other classes.

On the basis of the above description, it is necessary to turn single-task learning into multi-task learning in terms of both classification and feature selection. In future, we will investigate the use of fuzzy approaches towards identifying the relationships between different classes, based on the membership degree values of the same instances to these classes. It is also worthy of future research in the context of multi-granularity learning, especially when there are vertical relationships between classes, such as super-classes and sub-classes. For example, in the context of image processing, features could be extracted from different target regions and the classes (corresponding to the regions) could be located at different levels of granularity. In this case, both feature selection and classification need to be done in the setting of multi-granularity learning. In terms of feature selection, we will also compare the Prism algorithm with traditional approaches (filter and wrapper), in terms of their impact on the performance of popular learning algorithms. It is also worth to investigate the use of genetic algorithms (Chen and Chung 2006), parallelized genetic ant colony systems (Chen and Chien 2011), particle swarm optimization algorithms (Chen and Kao 2013) and parallel cat swarm optimization algorithms (Tsai et al. 2008, 2012), towards finding an optimal set of features.

Acknowledgements The authors acknowledge support for the research reported in this paper through the Research Development Fund at the University of Portsmouth and support from the China Scholarship Council and the Natural Science Foundation of Hebei Province, China (No. F2016203211).

References

Altrabsheh N, Cocea M, Fallahkhair S (2015) Predicting students' emotions using machine learning techniques. Springer, Berlin, pp 537–540

Antonelli M, Ducange P, Lazzerini B, Marcelloni F (2016) Multi-objective evolutionary design of granular rule-based classifiers. Granul Comput 1(1):37–58

Barber D (2012) Bayesian reasoning and machine learning. Cambridge University Press, Cambridge

Bergadano F, Cutello V (1993) Learning membership functions. In: European conference on symbolic and quantitative approaches to reasoning and uncertainty. Granada, pp 25–32

Berthold MR (2003) Mixed fuzzy rule formation. Int J Approx Reason 32:67–84

Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. Pattern Recognit 37:1757–1771

Cendrowska J (1987) Prism: an algorithm for inducing modular rules. Int J Man Mach Stud 27:349–370

Chatterjee K, Kar S (2017) Unified granular-number-based AHP-VIKOR multi-criteria decision framework. Granul Comput 2(3):199–221

Chen S-M (1996) A fuzzy reasoning approach for rule-based systems based on fuzzy logics. IEEE Trans Syst Man Cybern Part B Cybern 26(5):769–778

Chen S-M, Chang T-H (2001) Finding multiple possible critical paths using fuzzy pert. IEEE Trans Syst Man Cybern Part B Cybern 31(6):930–937

Chen S-M, Chen C-D (2011) Handling forecasting problems based on high-order fuzzy logical relationships. Expert Syst Appl 38(4):3857–3864

Chen S-M, Chien C-Y (2011) Parallelized genetic ant colony systems for solving the traveling salesman problem. Expert Syst Appl 38(4):3873–3883

Chen S-M, Chung N-Y (2006) Forecasting enrollments using high-order fuzzy time series and genetic algorithms. Int J Inf Manag Sci 17(3):1–17

Chen S-M, Kao P-Y (2013) Taiex forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines. Inf Sci 247:62–71

Chen S-M, Lee L-W (2010) Fuzzy decision-making based on likelihood-based comparison relations. IEEE Trans Fuzzy Syst 18(3):613–628

Chen S-M, Wang N-Y, Pan J-S (2009) Forecasting enrollments using automatic clustering techniques and fuzzy logical relationships. Expert Syst Appl 36(8):11070–11076

Dash M, Liu H (1997) Feature selection for classification. Intell Data Anal 1:131–156

David AB (2005) Real world data sets. https://www.cs.waikato.ac.nz/ml/weka/datasets.html. Accessed 6 Oct 2017

Dubois D, Prade H (2016) Bridging gaps between several forms of granular computing. Granul Comput 1(2):115–126

Furnkranz J (1999) Separate-and-conquer rule learning. Artif Intell Rev 13:3–54

Guyon I (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182

Horng Y-J, Chen S-M, Chang Y-C, Lee C-H (2005) A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. IEEE Trans Fuzzy Syst 13(2):216–228

Hu H, Shi Z (2009) Machine learning as granular computing. In: IEEE international conference on granular computing, Nanchang, pp 229–234

Jefferson C, Liu H, Cocea M (2017) Fuzzy approach for sentiment analysis. In: IEEE international conference on fuzzy systems, Naples

Kreinovich V (2016) Solving equations (and systems of equations) under uncertainty: how different practical problems lead to different mathematical and computational formulations. Granul Comput 1(3):171–179

Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86

Langley P (1994) Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall symposium on relevance. AAAI Press, Washington DC, pp 127–131

Lethbridge TC, Laganire R (2005) Object oriented software engineering: practical software development using UML and Java (2nd). McGraw-Hill Education, Maidenhead

Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed 4 Oct 2017

Liu H, Cocea M (2017a) Fuzzy information granulation towards interpretable sentiment analysis. Granul Comput 2(4):289–302

Liu H, Cocea M (2017b) Fuzzy rule based systems for interpretable sentiment analysis. In: International conference on advanced computational intelligence. Doha, pp 129–136

Liu H, Cocea M (2017c) Granular computing based approach for classification towards reduction of bias in ensemble learning. Granul Comput 2(3):131–139

Liu H, Cocea M (2018) Granular computing based machine learning: a big data processing approach. Springer, Berlin

Liu H, Gegov A (2015) Collaborative decision making by ensemble rule based classification systems. Springer, Switzerland, pp 245–264

Liu H, Gegov A, Cocea M (2016a) Generation of classification rules. In: Rule based systems for big data: a machine learning approach, vol 13, pp 29–42

Liu H, Cocea M, Gegov A (2016b) Interpretability of computational models for sentiment analysis. In: Pedrycz W, Chen SM (eds) Sentiment analysis and ontology engineering: an environment of computational intelligence, vol 639, pp 199–220

Liu H, Cocea M, Ding W (2017a) Decision tree learning based feature evaluation and selection for image classification. In: International conference on machine learning and cybernetics, Ningbo

Liu H, Cocea M, Mohasseb A, Bader M (2017b) Transformation of discriminative single-task classification into generative multi-task classification in machine learning context. In: International conference on advanced computational intelligence, Doha, pp 66–73

Liu P, You X (2017) Probabilistic linguistic TODIM approach for multiple attribute decision-making. Granul Comput 2(4):332–342

Livi L, Sadeghian A (2016) Granular computing, computational intelligence, and the analysis of non-geometric input spaces. Granul Comput 1(1):13–20

Mamdani E, Assilian S (1999) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Hum Comput Stud 51(2):135–147

Min F, Xu J (2016) Semi-greedy heuristics for feature selection with test cost constraints. Granul Comput 1(3):199–211

Montalto P, Aliotta M, Cannata A, Cassisi C, Pulvirenti A (2012) Similarity measures and dimensionality reduction techniques for time series data mining. In: Karahoca A (ed) Advances in data mining knowledge discovery and applications, InTech, pp 71–96

Pedrycz W, Chen S-M (2011) Granular computing and intelligent systems: design with information granules of higher order and higher type. Springer, Heidelberg

Pedrycz W, Chen S-M (2015a) Granular computing and decision-making: interactive and iterative approaches. Springer, Heidelberg

Pedrycz W, Chen S-M (2015b) Information granularity, big data, and computational intelligence. Springer, Heidelberg

Pedrycz W, Chen S-M (2016) Sentiment analysis and ontology engineering: an environment of computational intelligence. Springer, Heidelberg

Peters G, Weber R (2016) DCC: a framework for dynamic granular clustering. Granul Comput 1(1):1–11

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423

Skowron A, Jankowski A, Dutta S (2016) Interactive granular computing. Granul Comput 1(2):95–113

Tan P-N, Steinbach M, Kumar V (2005) Introduction to data mining. Addison-Wesley Longman Publishing Co. Inc., Boston

Teng Z, Ren F, Kuroiwa S (2007) Emotion recognition from text based on the rough set theory and the support vector machines. In: International conference on natural language processing and knowledge engineering, Beijing, pp 36–41

Tsai PW, Pan JS, Chen SM, Liao BY, Hao SP (2008) Parallel cat swarm optimization. In: Proceedings of the 2008 international conference on machine learning and cybernetics, Kunming, vol 6, pp 3328–3333

Tsai P-W, Pan J-S, Chen S-M, Liao B-Y (2012) Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst Appl 39(7):6309–6319

Tsoumakas G, Katakis I (2007) Multi-label classification: an overview. Int J Data Warehous Min 3(3):1–13

Tsoumakas G, Katakis I, Vlahavas I (2010) Mining multi-label data. In: Data mining and knowledge discovery handbook. Springer, Berlin, pp 667–685

Wang L-X, Mendel JM (1992) Generating fuzzy rules by learning from examples. IEEE Trans Syst Man Cybern 22(6):1414–1427

Wilke G, Portmann E (2016) Granular computing as a basis of human-data interaction: a cognitive cities use case. Granul Comput 1(3):181–197

Wu J, Smith WA, Hancock ER (2011) Gender discriminating models from facial surface normals. Pattern Recognit 44(12):2871–2886

Xu Z, Wang H (2016) Managing multi-granularity linguistic information in qualitative group decision making: an overview. Granul Comput 1(1):21–35

Yao J (2005a) Information granulation and granular relationships. In: IEEE international conference on granular computing, Beijing, pp 326–329

Yao Y (2005b) Perspectives of granular computing. In: Proceedings of 2005 IEEE international conference on granular computing, Beijing, pp 85–90

Yu L, Liu H (2003) Feature selection for high-dimensional data: a fast correlation-based filter solution. In: Proceedings of the 20th international conference on machine learning (ICML-03), Washington DC, pp 856–863

Zadeh L (1965) Fuzzy sets. Inf Control 8(3):338–353

Zhang M-L, Zhou Z-H (2014) A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng 26(8):1819–1837